MD*Book and XQC/XQS - an Architecture for Reproducible Research

Sigbert Klinke, Heiko Lehmann,

CASE, Humboldt-Universität zu Berlin, Spandauer Strasse 1, 10178 Berlin, Germany

Abstract. Statistical software has also become an important part of scientific research that is reflected in the publications of the research results. Publishing a mathematical theorem requires also the publication of the proof of this theorem. The result of a computation can be seen as the equivalent of a mathematical theorem. Reproducibility of published results allows to fulfill this demand – offers the possibility to proof computational results.

Keywords. Reproducible Research, MD*Book, XploRe Quantlet Client/Server, e-Books, Java

1 Introduction

Nowadays there hardly exists a field where research can be carried out without the (mostly extensive) use of computers. Models that are becoming more and more complex and huge datasets are characteristics of today's science. The financial researcher uses mathematical and statistical models to predict market behavior, the social scientist for research about the deployment of the birthrate in Germany or the physicist for modeling the influence the temperature has on the conductance of copper.

When publishing results of statistical research, it is desirable for an interested reader to be able to verify and regenerate these results. Even more desirable to an interested researcher is to be able to inspect the source code, modify it and produce variations of the results. Buckheit and Donoho (1995) outline the topic of Reproducible Research - "An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures." They propose to publish research papers as electronic books and include the software environment the results where generated with to make them interactively accessible.

Reasons for making research reproducible can be very manifold.

Buckheit and Donoho (1995) probably thought of long nights of work while describing one possible reason: Writing an article that contains lots of figures as result of research often takes quite some time. During this period of time hundreds of figures are usually generated – varying algorithms and parameters. At the end the question might arise – *which* figures were the final versions that should be used for the article? Taking "the

nicest looking figures" would not be the right choice. Using the figures that were generated using the final algorithms and parameters exactly described in the article would be the way to go. They might not be the best looking ones but the true results of the research. Publishing reproducible figures forces the author to use the right graphics, the right results, since every reader is actually be able to regenerate this figure.

Consider yourself reading your own paper one, two or even more years later. Would you be able to generate the same picture as in your paper again? If - by accident - you do not have the original program coding hidden somewhere it is going to be not that easy. Even though you would still have the software coding the according environment might have been changed in the meantime making it impossible to reproduce your results. Using the new media like CD-ROM makes it possible to not just publish the written paper but also the software or parts of it that have been used for research.

Thinking of a young student who begins to do scientific research leads to another reason. Very often students are facing the problem to find the right start into their topic. Although they are usually able to access tons of literature sources most of them are "limited" to just explaining algorithms. Rarely the used parameters are included in published papers. In order to use those algorithms as a starting point the student tries to reproduce the previous result. This means programming the entire logic again to use it as basis for further study. Reproducible research can help to make the entry in the world of scientific research much easier for young students. They could actually be able to try the programs, change and vary parameters or even extend existing programs during their study. On the other hand being able to publish results within interactive documents allow those young researchers to easy share their results and knowledge with other scientists.

To summarize possible reasons for making research reproducible – it will

- make research more accurate and more credible, since other researchers are able to discover any errors and to validate the results
- save time for other researchers working on the same or a similar task, since their research can be build up on validated and extendible results
- most likely help young students beginning their research

Presentation of research results is usually limited to formulas and graphics, their description and additional explanations. Main reasons for this limitation are the austerities given by the medium that is most often used - common paper. But the quite fast growth of the internet and dropping prices for electronic media like CD-ROM and DVD-ROM have started to change the dominance and autarchy of paper. These Media allow for presentations that exceed the message a simple graphic can deliver.

2 Types of Presentation

How can results of a scientific research be presented to the audience? What are the advantages and disadvantages of the different ways? The following part tries to find answers to these questions. To illustrate the different forms of representing scientific content a very simple example from basics statistics will be used – analyzing the coherence for decathletes between the results of 100 m and long jump. As simple as this example my seem it often works as a basis for further research and more complex algorithms. The more complex the methods and algorithms of a mathematical or statistical research get the more effort is necessary to prepare and to present the results of this research. The dataset – decathlon.dat – we use for our example can be found in appendix A.

We define different level or stages respectively for presenting statistical and mathematical results within research documents:

Text only

The simplest form of presenting results of scientific research is a pure textual description. This form of presentation is limited to the use of just algorithms, formulas and explanations of coherences. A disadvantage of this method is yet the presupposed ability of any reader to abstract. This ability is often needed in order to understand the content, to put oneself in the authors position. Using the decathlon example we get the following results:

- Correlation coefficient: -0.69
- Regression function: long jump = 17.1825 – 0.8988 * 100 m
- Coefficient of determination: 0.48

The data itself can be presented within a table – see table 1.

Decathlete	A	B	C	…
100 m Sprint	11.25	10.87	11.18	...
Weitsprung	7.43	7.45	7.44	...

Table 1: Results of a decathlon

What are the results of this simple research? Interpreting the calculated results leads to the conclusion, that there seems to be a coherence between the two sports. But without taking a closer look at the data itself and/or calculating additional coefficients a "final" conclusion is hardly possible. The dataset could for example contain outliers that influence the results. It takes more numbers and explanations to get an image of the actual coherence. Many readers of those kind of publications try to transform those pure numbers into graphics using their imagination. With this simple example this might still be possible, but imagine the presentation of ???.

Graphical representation

The use of graphics for presenting the results of scientific research is an extension of the method just described above. Graphics are meant to help the author as well as the reader of publications to recognize and understand coherences in datasets and to get an image of presented algorithms. Figure 1 shows a scatter plot of our decathlon example used in the previous section. In addition the calculated regression line printed within the plot. This figure simplifies an interpretation of the calculated coefficients and makes it easier to identify a statistical coherence between 100 m and long jump.

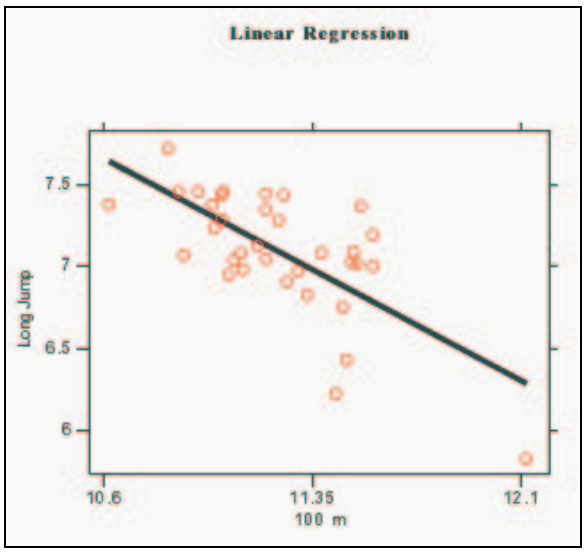

Figure 1: Scatter plot and regression line

The two forms of representing scientific results have one thing in common. It is hardly traceable where the calculated results and graphics come from. Are they an outcome of a mathematical or statistical software environment or is e.g. the presented graphic the result of a graphical program. We absolutely do not want to doubt publications without the possibility to reproduce results, but nevertheless the reader has to believe in the authors calculations. An additional publication of used datasets, source code of programs used for calculation of results and generation of graphics and information about the used software environment are basic approaches to give an interested reader at least the chance to verify and reproduce the results. The pure source code is only useful if the reader has access to the same software, libraries and methods the author has used. But in many cases this might not be that easy.

Reproducibility of calculated results and graphics

What does reproducibility or reproducible research respectively mean? Gentleman and Lang (2003) define: "Reproducible research is reproducible in the sense that the author has provided sufficient detail (in the form of code and data) for a reader to reproduce the details of the authors presentation." Following to this definition reproducibility would be given by adding all used data and the used source code to a publication. As already

mentioned above this information do not automatically enable the interested reader to reproduce the research. Without access to the statistical software environment the author has used it is hardly possible to reproduce any result. In addition software is often subject to short life cycles in terms of versioning. A year can thus be a long time, making the source code worthless. For solving this kind of problem Buckheit and Donoho (1995) go one step further – "When we publish articles containing figures which were generated by computer, we also publish the complete software environment which generates the figures." Both citations can be combined to a more abstract definition – reproducibility means the possibility for the reader of a document to re-compute results or re-build figures respectively.

Figure 1 was generated using the software environment XploRe. For our example reproducibility could mean to publish the source code of the XploRe Quantlet we used. In this case the interested reader needs to have access to this software environment in order to reproduce the figure. Nowadays more and more software developers offer access to their software via web interface. Attaching the source code to an article is the easiest way to realize reproducibility. The advance of this approach is the independence of media other than pure paper. There is no need for additional software that would have to be published as well. The drawback this approach is the limitation, that the reader cannot regenerate the figure right away but has to "break out" of the article.

If the article is published e.g. via Internet or on CD-ROM another way for making our figure reproducible could be used. Clicking on a link within the document (pdf or html) would open an additional window on the screen where the same figure as published in the article is re-generated. Advance of this approach is the possibility to reproduce results without actually having to leave the document itself. It could even offer more functionality than just showing a figure – like in our case additional functionality can be offered to the reader. This functionality could include showing the coordinates of a data point, printing functionality or – if a 3-dimensional plot has been generated – offering the possibility to rotate the plot. Drawback of this approach is the dependence of interactive media.

Reproducibility and interactivity regarding data

Interactivity regarding data represents an extension of the simpler form reproducibility described above. With this kind of reproducibility readers are not limited to use only the data published by the author, but they get the chance to apply the algorithms on their own data. In difference to the previously discussed approaches interactivity implies the usage of media other than paper. Whereas a simple reproducibility would be given by publishing the source code for offering interactivity the software environment needs to be published as well. Internet, CD-ROM or DVD-ROM recommend themselves for publishing. This does not mean to replace printed-paper. This possibility should rather be seen as a supplement than a replacement.

Being able to change or vary data enables the reader of this kind of publication – the reader actually becomes a user – to really verify research results. Playing "What if …" scenarios, for example eliminating outliers, helps to understand presented theories. An additional advantage is the possibility to apply new published algorithms to the reader's own data.

Reproducibility and interactivity regarding data and statistical program

Even more useful for many readers would be the possibility to have access to the underlying program that has generated the calculations and graphics. This feature does not just enable the user to validate the authors results but the source code can be used as a basis for further research.

For this form of representation the same drawbacks as for the previous methods apply – its usage presupposes a publication on interactive media.

3 Making research reproducible

Our approach presented in the following sections tries to allow for reproducibility of scientific results as well as for interactivity in scientific publications. Aim of this approach is to combine

- fundamentals of a research (the used data)
- results (theorems, algorithms, graphical and non-graphical outcome)
- final conclusions

within an interactive document.

3.1 The XQC/XQS Model

The XploRe Quantlet Client/Server model represents the part, that enables reproducibility of results and interactivity (see Kleinow and Lehmann 2001) It consists of the components server, middleware and client.

The **XploRe Quantlet Server** (XQS) is offering services to one or more client(s). Based on the statistical software environment XploRe with its high-level statistical programming language the server represents the computing engine of the model. Running on a remote computer the XQS can offer a magnitude of computer power, which many users would not be able to access in other ways. Having access to the method- and database the XQS and the method- and database respectively is easily extendible by new statistical methods via XploRe programs (Quantlets) as well as native code methods, e.g. -dll and -so. For server side communication purposes the middleware MD*Serv is attached to the XQS. The Communication between MD*Serv and XploRe server is

realized via standard I/O streams – the middleware reads from the server's standard input and writes to its standard output.

Middleware – running on server's side as well as on client's side enables the communication between client and server. An especially for this architecture developed protocol (MD*Crypt) is used for this purpose (see Feuerhake).

The **XploRe Quantlet Client** (XQC) represents the Front-End of the architecture. The client is fully programmed in Java2. Due to this feature a platform independent and universal usage is possible. The XQC can be used running as an application as well as running as an applet within a web browser.

The XQC does not just work in the common way most clients do – it does not only react to commands given by the actual user. Instead it is possible to setup the client to be started in certain ways defined by the author. For this purpose a special property file is used. The property file itself is a simple ASCII file containing commands. These commands allow starting the XQC with

- '*ExecuteCommand*' – executing a command
- '*ExecuteProgram*' – executing a XploRe Quantlet
- '*OpenData*' – opening a data set
- '*OpenInEditor*' – opening a XploRe Quantlet.

Using the commands stated above leads to different levels of reproducibility or interactivity respectively.

Reproducibility of calculated results and graphics is the simplest form of reproducibility. This feature can be realized using the command:

ExecuteProgram = file:///C:/.../QRegression01.xpl

In this case the XQC starts and executes the stated XploRe Quantlet right away but without showing the coding. As a result a graphic is generated that is identical to the graphic in the written document. Even though it is the simplest form of reproducibility the graphic offers some features via a context menu the pure written document hardly can – coordinates of the data can be shown, for 3D plots rotation via cursor keys or mouse is possible.

If parameters have been used for the computation of results or the generation of graphics the author has the choice to extend this form of reproducibility by interactive components the XploRe programming language offers. Using XploRe's *selectitem* or *readvalue* for the Quantlet the author can give the possibility to the reader to change and adjust parameters, to influence results and graphics of the computation (see XploRe's APSS - http://www.i-xplore.de/help/_Xpl_Start.html - for more information).

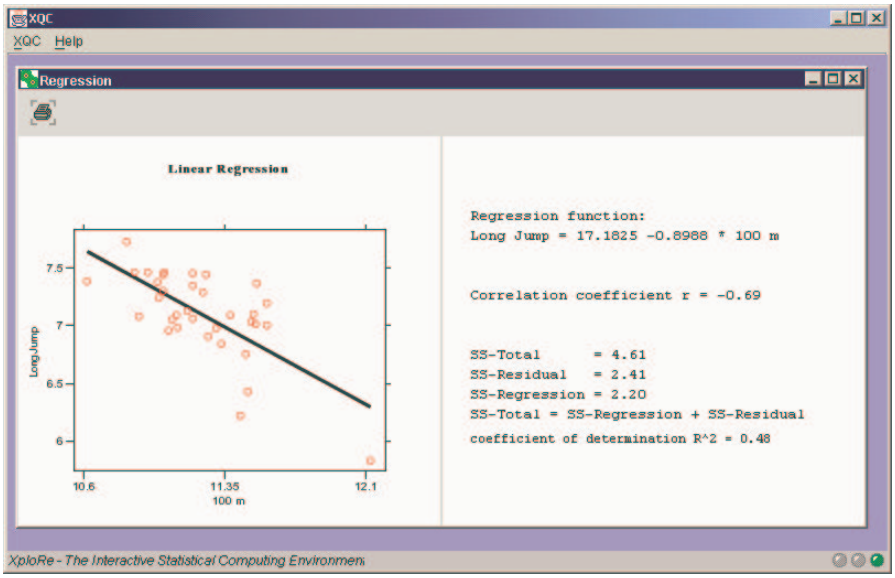

Figure 2: Reproducible research

For realizing **reproducibility and interactivity regarding data** the following parameters need to be defined in the property file:

OpenData = *XQCROOT/decathlon.dat*
ShowMethodTree = *yes*
MethodTreeIniFile = *xqc_regression.ini*
MethodPath = *XQCROOT/xqc_quantlets/*

As described before path statements can be maintained as absolute paths – locally (file:///…) or URL (http://…) – as well as relative to the directory the XQC has been started in (XQCROOT/…). A second property file – in this case xqc_regression.ini – defines the XploRe Quantlet – in this case QRegression_02.xpl – that is used on the data. The Quantlet to be used is stored in a subdirectory '*xqc_quantlets/*' relative to the directory the XQC has been started in.

Content of the property file *xqc_regression.ini*:

Child_1 = *QRegression02|Regression*

The first part '*QRegression02*' defines the name of the Quantlet and its main procedure. The second part '*Regression*' represents the name that is shown within the method tree.

Using the parameters stated above the XQC starts with opening the stated data set (*decathlon.dat*) and a method tree containing the predefined method (*Regression*).

With this form of representation the interested reader is not just able to verify the data but also to change data and explore how this affects the regression coefficients and the

graphic. Going one step further the reader would even be able to use his/her own data for computation.

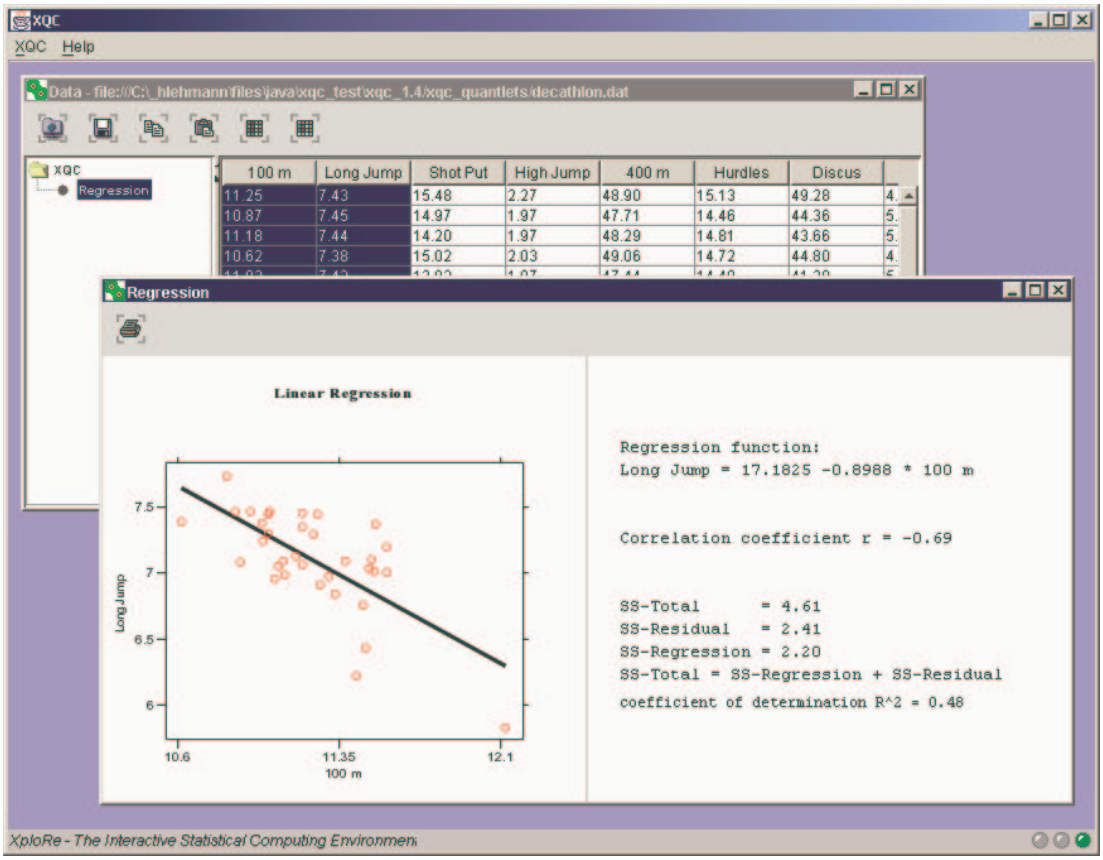

Figure 3: Interactivity regarding data

Interactivity regarding data and statistical program

An extension of the approach described above is the possibility to edit the actual XploRe Quantlet that has computed the results. Using the parameter

OpenInEditor = file:///C:/.../test1.xpl

enables this feature. Right after the XQC has been started an editor window opens containing the code. The interested reader can explore, edit and execute the program. If the parameter *'OpenInEditor'* is used in addition to the parameter *'OpenData'* the user can edit both – program and data. Changed data can be uploaded to the server and used within the Quantlet. Figure 4 shows a screenshot of the XQC offering interactivity regarding data and program.

With this form of reproducibility the interested reader has almost all possibilities to verify research results. The reader is not limited to the program and data given by the author but can use his/her own data and program extensions. This can thus be a good start for an

ongoing research in the same topic. Researches do not have to start from the very beginning again.

As seen above using the XQC's functionality to make research reproducible does not require any Java programming skills for adjusting the client. Every XploRe Quantlet the author has used for his/her research can easily be made available for reproduction by just maintaining some parameters in ASCII style property files. The final integration into research documents will be explained in the next chapter.

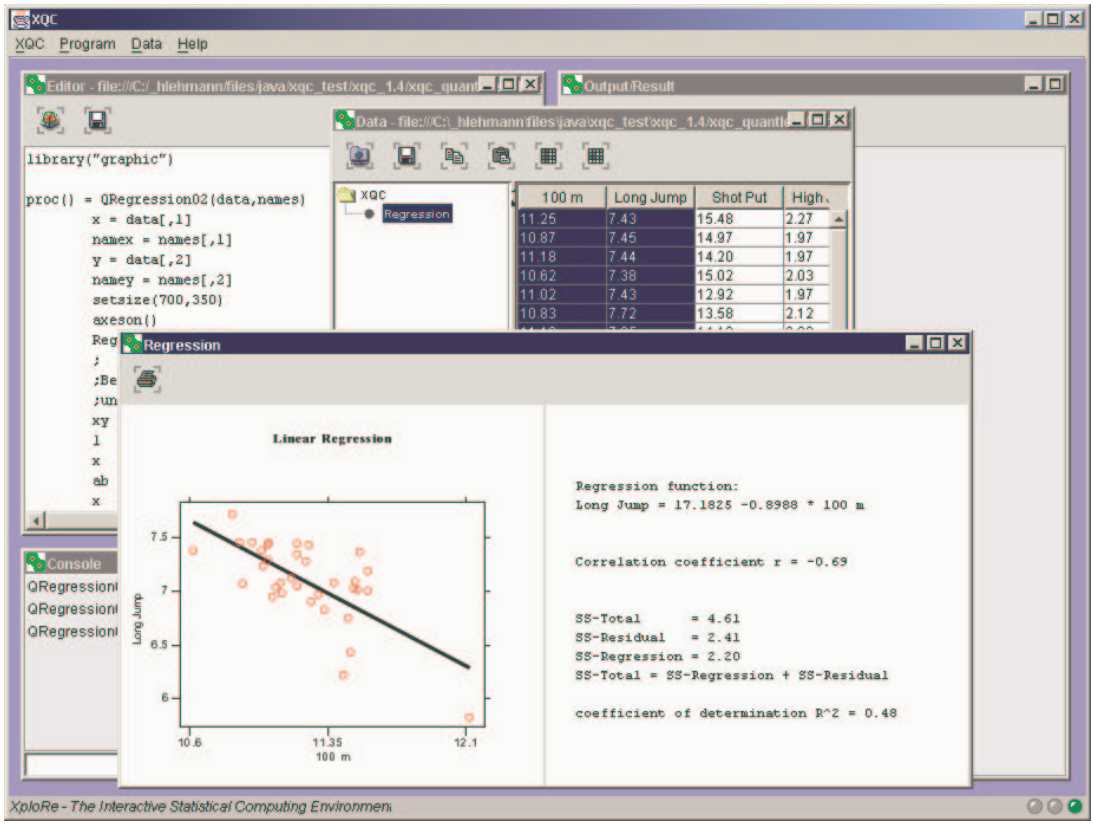

Figure 4: Interactivity regarding data and program

3.2 MD*Book

The Client/Server structure of XploRe allowed us to integrate XploRe programs (Quantlets) into a web page. We call this methodology the “Golden solution”. We see in Figure 5 a short description what the Quantlet “XLGsmoo09”is doing. We have a link to the source code (for downloading) and we can inspect the code immediately.

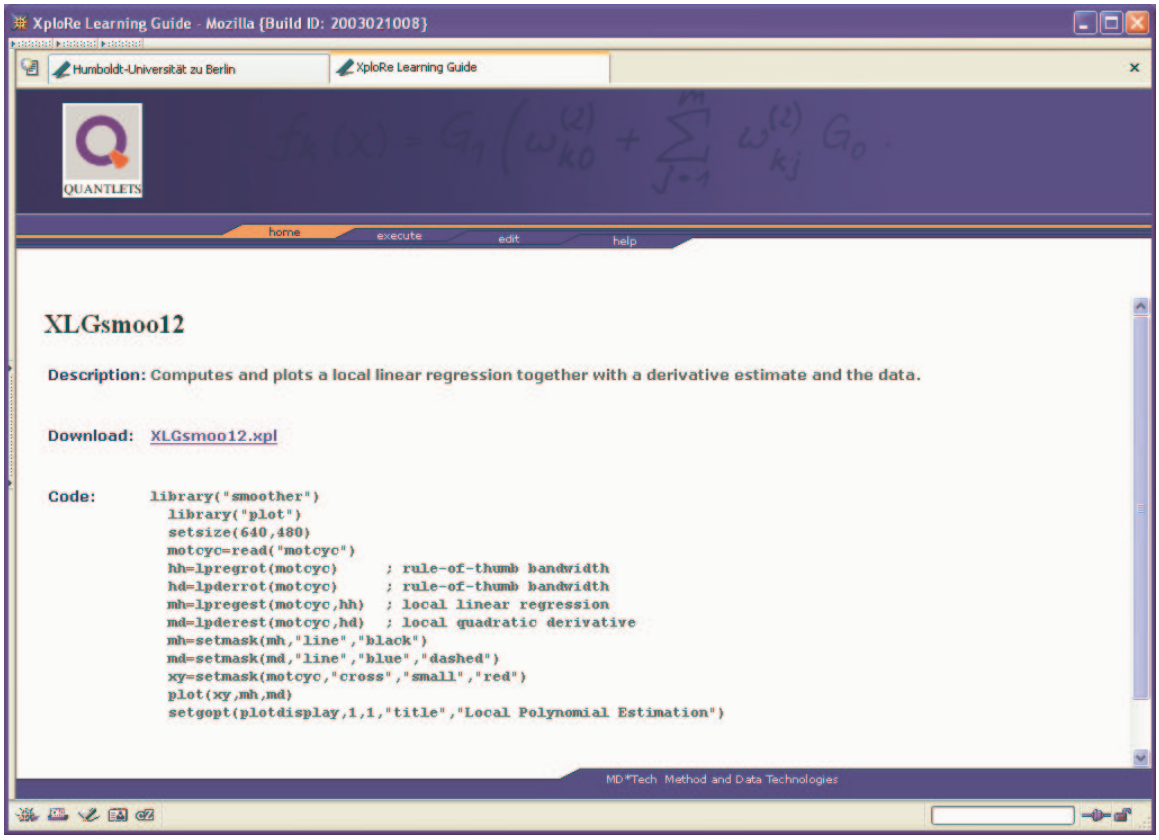

Figure 5: Example of Reproducibility generated using MD*Book

The tab strips “execute” and “edit” allow different access to the source code. An inexperienced user will choose “execute” which will simply execute the code. An experienced user might choose “edit” which first shows the source code in the XploRe editor. Now the user can modify the code or run it (see Figure 6).

To ensure that the code on the web page and the source code at the server are the same, was one of the reasons to develop MD*Book. As in Witzel and Klinke (2002) described the option –xpl generates from an XploRe Quantlet the necessary web pages.

The other root for developing MD*Book was that we were able to generate PostScript, PDF and HTML pages from LaTeX sources. Unfortunately a lot of different tools like latex, dvips, pdflatex and latex2html were used.

The bundling of all these tools including the generation of the “Golden solution” is the MD*Book project (see Figure 7).

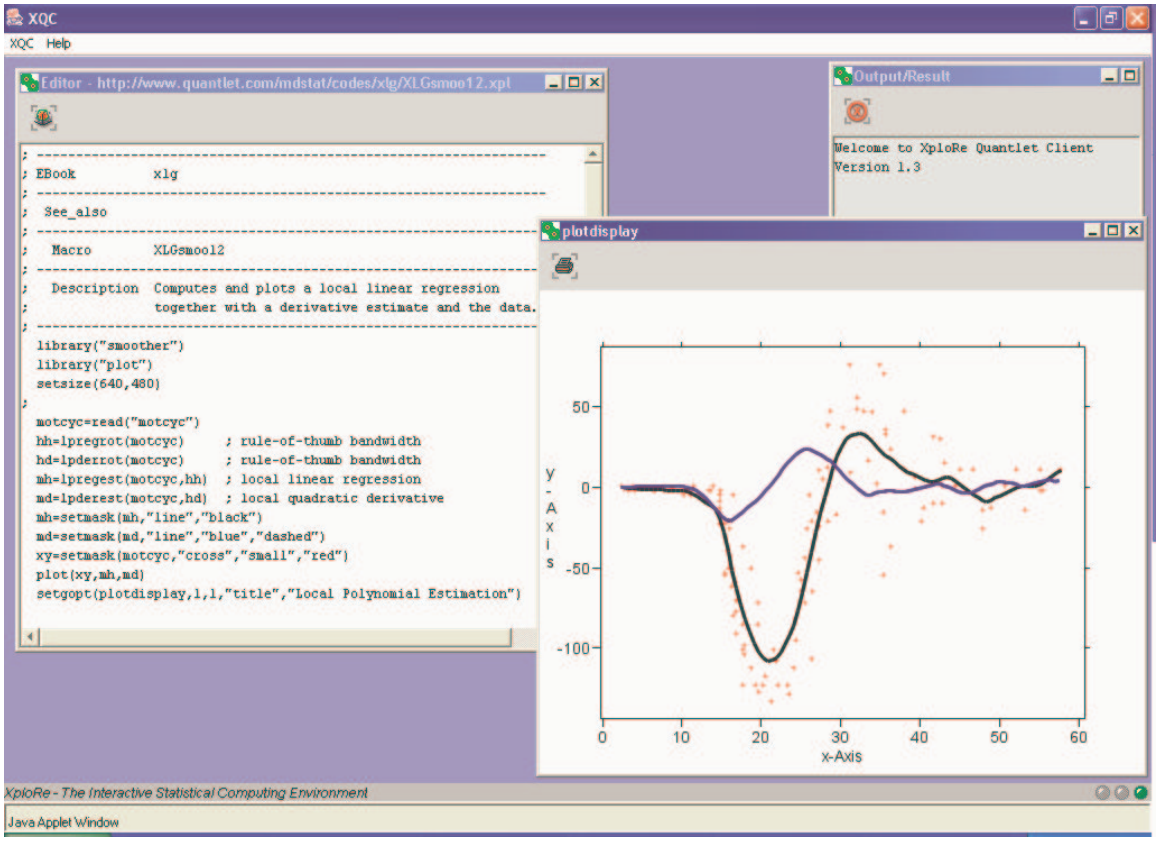

Figure 6: Interactive example

To integrate for a figure the generating program requires only the integration of a link to the appropriate web page with the program.

```
\begin{verbatim}
  motcyc=read("motcyc")
  hh=lpregrot(motcyc)     ; rule-of-thumb bandwidth
  hd=lpderrot(motcyc)     ; rule-of-thumb bandwidth
  mh=lpregest(motcyc,hh)  ; local linear regression
  md=lpderest(motcyc,hd)  ; local quadratic derivative
  mh=setmask(mh,"line","black")
  md=setmask(md,"line","blue","dashed")
  xy=setmask(motcyc,"cross","small","red")
  plot(xy,mh,md)
  setgopt(plotdisplay,1,1,"title","Local Polynomial Estimation")
\end{verbatim}
\clink{XLGsmoo12}

\begin{figure}[htb]
  \begin{center}
    \ineps{0.425}{smootherlld}
    \caption{Local linear regression (solid), derivative (dashed)
      estimate and data.%
      \label{smoo_regld}}
  \end{center}
```

```
\end{figure}
```

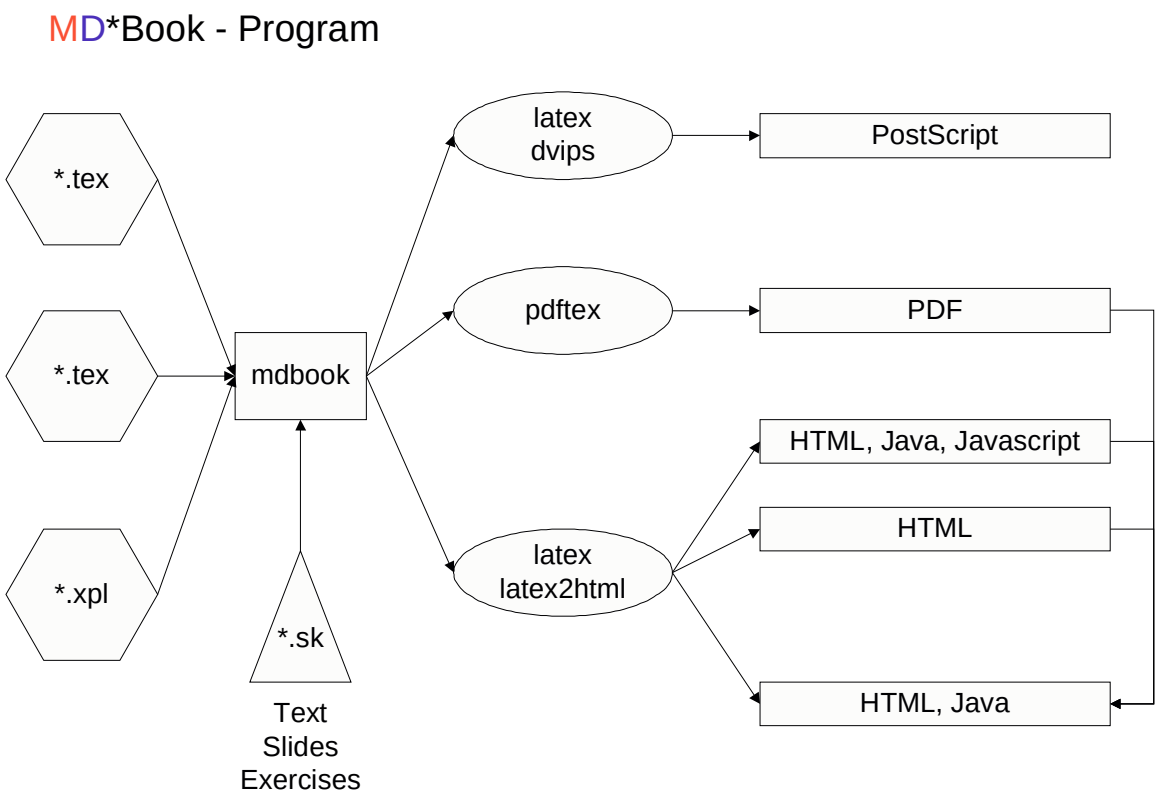

Figure 7: MD*Book project

4 References

Buckheit, J., Donoho, D. (1995), WaveLab and Reproducible Research, in Wavelets and Statistics, A. Antoniadis, ed., Springer-Verlag, Berlin, New York.

Clearbout

Feuerhake, J., (2001), MD*CRYPT – the XQS/XQC protocol, available online at http://www.md-crypt.com

Gentleman and Lang (2003)

Härdle, W., Klinke, S., Müller, M. (1999), XploRe -The Statistical Computing Environment, Springer-Verlag, New York.

Kleinow, T., Lehmann, H. (2001), Client/Server based Statistical Computing, Proc. of the ISM symposium "Statistical software in the Internet age", 1-8, Institute of Statistical Mathematics, Tokyo.

Leisch

Witzel, R., Klinke, S. (2002), MD*Book online & e-stat: Generating e-stat Modules from LaTeX, Proceedings in Computational Statistics 2002, p.449-454.